Graph Meets LLMs: Towards Large Graph Models

Ziwei Zhang, Haoyang Li, Zeyang Zhang, Yijian Qin, Xin Wang, Wenwu Zhu
Department of Computer Science and Technology, Tsinghua University. Beijing, China, 100084
{zwzhang,xin_wang,wwzhu}@tsinghua.edu.cn
{lihy18,zy-zhang20,qinyj19}@mails.tsinghua.edu.cn

Abstract

Large models have emerged as the most recent groundbreaking achievements in artificial intelligence, and particularly machine learning. However, when it comes to graphs, large models have not achieved the same level of success as in other fields, such as natural language processing and computer vision. In order to promote applying large models for graphs forward, we present a perspective paper to discuss the challenges and opportunities associated with developing large graph models[1]. First, we discuss the desired characteristics of large graph models. Then, we present detailed discussions from three key perspectives: representation basis, graph data, and graph models. In each category, we provide a brief overview of recent advances and highlight the remaining challenges together with our visions. Finally, we discuss valuable applications of large graph models. We believe this perspective can encourage further investigations into large graph models, ultimately pushing us one step closer towards artificial general intelligence (AGI). We are the first to comprehensively study large graph models, to the best of our knowledge.

1 Introduction

In recent years, there has been a growing interesting in large models for both research and practical applications. Large models have been particularly revolutionary in fields such as natural language processing (NLP) [1, 2, 3] and computer vision (CV) [4, 5, 6], where pre-training extremely large models on large-scale unlabeled data has yielded significant breakthroughs. However, graphs, which are commonly used to represent relationships between entities in various domains such as social networks, molecule graphs, and transportation networks, have not yet seen the same level of success with large models as other domains. In this paper, we present a perspective about the challenges and opportunities associated with developing large graph models. First, we introduce large graph models and outline four key desired characteristics, including graph models with scaling laws, graph foundation model, in-context graph understanding and processing abilities, and versatile graph reasoning capabilities. Then, we offer detailed perspectives from three aspects: (1) For graph representation basis, we discuss graph domains and transferability, as well as the alignment of graphs with natural languages. Our key takeaway is the significance of identifying a suitable and unified representation basis that spans diverse graph domains, which serves as a fundamental step towards constructing effective large graph models; (2) For graph data, we summarize and compare the existing graph datasets with other domains, and highlight that the availability of more large-scale high-quality graph data is resource-intensive yet indispensable; (3) For models, we systematically discuss backbone architectures, including graph neural networks and graph Transformers, as well as pre-training and post-processing techniques, such as prompting, parameter-efficient fine-tuning, and model compression. We also discuss LLMs as graph models, which is a newly trending direction. Finally, we discuss the significant impact that large graph models can have on various graph applications, including recommendation systems, knowledge graphs, molecules, finance, code

[1] We maintain a curated paper list at `https://github.com/THUMNLab/awesome-large-graph-model`.

NeurIPS 2023 New Frontiers in Graph Learning Workshop (NeurIPS GLFrontiers 2023).

and program, and urban computing and transportation. We hope that our paper can inspire further research into large graph models[2].

2 Desired Characteristics of Large Graph Models

Similar to large language models (LLMs) [3], a large graph model can be characterized as a graph model with a vast number of parameters which empower it with abilities that are substantially more powerful than smaller models, thereby promoting the understanding, analyses, and processing of graph-related tasks. Apart from having numerous parameters, we summarize the key desired characteristics of an ideal large graph model from the following perspectives. An illustration of these characteristics is provided in Appendix A.

1. **Graph models with scaling laws**: The scaling laws indicate an empirical phenomenon where the performance of LLMs continues to improve as the model size, dataset size, and training computation increase [16]. This phenomenon offers a clear direction for enhancing performance and empowering the model to capture complex patterns and relationships within graph data. By emulating the success of LLMs [17], a large graph model is expected to exhibit emergent abilities that smaller models lack. However, accomplishing this objective in large graph models is highly non-trivial, with difficulties span from collecting more graph data to solving technical problems such as addressing the over-smoothing and over-squashing problem of graph neural networks, along with engineering and system challenges.
2. **Graph foundation model**: A large graph model holds greater value when it can serve as a graph foundation model, i.e., capable of handling different graph tasks across various domains. This requires the model to gain understandings of the inherent structural information and properties of graphs to be equipped with "commonsense knowledge" of graphs. The graph pre-training paradigm is a highly promising path to develop graph foundation models, as it can expose the model to large-scale unlabeled graph data and reduce the reliance on expensive and laborious collection of graph labels. Besides, a generative pre-training paradigm can potentially empower the model with the ability to generate graphs, thereby opening up possibilities for valuable applications like drug synthesis, code modeling, and network evolution analysis [18]. It is worthy clarifying that, since graphs serve as general data representations with extreme diversity, it is exceedingly challenging, if not unlikely, to develop a "universal graph model" for all graph domains. Therefore, multiple graph foundation models may be necessary for different "clusters of domains", which is somewhat different from LLMs or foundation models in computer vision.
3. **In-context graph understanding and processing abilities**: An effective large graph model is expected to comprehend graph contexts, including nodes, edges, subgraphs, and entire graphs, and process novel graph datasets and tasks during testing with minimum samples as well as without intensive model modifications or changes in the paradigm. This characteristic is also closed related to and can facilitate capabilities of few-shot/zero-shot graph learning [19], multi-task graph learning [20], and graph out-of-distribution generalization [21]. Moreover, these abilities are vital when the input graph data and task are different between the training and testing stages. In-context learning abilities can enable large graph models to leverage knowledge learned in the pre-training stage and quickly adapt to the testing stage with desired performance.
4. **Versatile graph reasoning capabilities**: Although graphs span diverse domains, there exist common and fundamental graph tasks. We generally refer to handling of these tasks as "graph reasoning". While there is no clear consensus on what these tasks are, some representative examples are provided as follows. Firstly, a large graph model should understand basic topological graph properties, such as graph sizes, node degrees, node connectivity, etc. These properties form the foundation for a deeper understanding of graph structures. Secondly, a large graph model should be able to flexibly and explicitly reason over multi-hop neighborhoods, enabling it to perform more sophisticated tasks. Such capabilities, akin to the chain-of-thought of LLMs [22] in principle, can also enhance transparency in the graph decision-making process and improve model explainability [23]. Lastly, besides local information, a large graph model should be able to understand and handle graph tasks that involve global properties and patterns, such as the centrality and position of nodes, overall properties of graphs, the evolution laws of dynamic graphs, etc.

[2]There are also works to use graphs to improve large language models, such as enhancing their reasoning ability [7, 8, 9, 10, 11, 12, 13] or using graphs as tools [14, 15], which is beyond the scope of this paper.

3 Graph Representation Basis

3.1 Graph Domains and Transferability

Large models, LLMs, serve as foundation models [24], as they can be adapted to a wide range of downstream tasks after being pre-trained. The remarkable ability of LLMs stems from the underlying assumption of the existence of a common representation basis for various NLP tasks. For instance, word tokens for natural language processing are universal and information-preserving data representations that do not rely on specific tasks. In contrast, graphs are general data structures that span a multitude of domains. Therefore, the raw input data, i.e., nodes and edges, may not always be the most suitable representation basis for handling all graph data. Nodes and edges in social networks, molecule graphs, and knowledge graphs, for instance, have distinct meanings with their unique feature and topological space. Thus, directly sharing information and transferring knowledge based on input graph data often poses significant challenges.

It is widely believed that there exist more high-level or abstract common graph patterns, which can be shared across different graphs and tasks within a certain domain. For example, many human interpretable patterns have been identified in classical network science [25], such as homophily, small-world phenomenon, power-law distribution of node degrees, etc. Nevertheless, even with these high-level shared knowledge, creating effective large models that can perform well across diverse graph domains is still non-trivial.

3.2 Aligning with Natural Languages

Another key competency of recent large models is their ability to interact with humans and follow instructions, as we are naturally capable of understanding languages and visual perceptions. In contrast, humans are less capable of handling graphs, especially more complex reasoning problems. As a result, communicating and instructing large models to behave for graph tasks the way we desire, especially using natural languages, is particularly challenging. We summarize three categories of strategies worth exploring to overcome this obstacle.

The first strategy is to align the representation basis of graphs and text through a large amount of paired data, similar to computer vision in principle. If successful, we will be able to interact with graph models using natural languages. For example, we can ask the model to generate molecule graphs with desired properties or ask the model to perform challenging graph reasoning tasks. Some initial attempts have been made for text-attributed graphs [26, 27], which serve as a good starting point. However, collecting such data for general graphs is much more costly and challenging than image-text pairs.

The second strategy is to transform graphs into natural languages, and then work solely in the language basis. Some initial attempts using this strategy have been developed, where graph structures are transformed into text representations, such as the adjacency list or the edge list, and inserted into LLMs as prompts. Then, natural languages are used to perform graph analytical tasks. We provide more detailed discussions in Section 5.4. However, directly transforming graph data and tasks into languages may lose the inner structure and inductive bias for graphs, resulting in unsatisfactory task performance. More delicate designs, such as effective prompts to convert graph structures and tasks into texts, are required to further advance this strategy.

The last category is to find other representation basis as a middle ground for different graph tasks and natural languages. The most straight-forward way is to use some hidden space of neural networks. However, it faces the challenge that deep neural networks are largely not explainable at the moment, not to mention that finding the desired shared hidden space can be frustratingly challenging. On the other hand, although humans are not capable of directly handling graph data, we can design appropriate algorithms to solve graph tasks, including many well-known algorithms in graph theory such as finding shortest paths, dynamic programming, etc. Therefore, if we can align the behavior of graph models with these algorithms, we can understand and control the behaviors of these models to a certain extent. Some efforts have been devoted in this direction, known as algorithmic reasoning [28], which we believe contains rich potentials.

In summary, finding the suitable representation basis, potentially aligning with natural languages, and unifying various graph tasks across different domains is one fundamental step towards building successful large graph models.

4 Graph Data

The success of big models is largely dependent on the availability of high-quality, large-scale datasets. For instance, GPT-3 was pre-trained on a corpus of approximately 500 billion tokens [1], while CLIP, a representative model that bridges natural language processing and computer vision, was trained on 400 million image-text pairs. It is reasonable to assume that even more data has been utilized in more recent large models, such as GPT-4 [29]. This massive amount of data for NLP and CV tasks is typically sourced from publicly accessible human-generated content, such as web pages in CommonCrawl or user-posted photos in social media, which are easily collected from the web.

In contrast, large-scale graph data is not as easily accessible. There are typically two scenarios for graph data: numerous small-scale graphs, such as molecules, or a single/few large-scale graphs, such as social networks or citation graphs. For example, Open Graph Benchmark [30], one of the most representative public benchmarks for graph machine learning, includes two large graph datasets: MAG240M, which contains a large citation graph with approximately 240 million nodes and 1.3 billion edges, and PCQM4M, which contains approximately 4 million molecules. However, their scale is considerably lower than the datasets used in NLP or CV. If we treat each node in MAG240M as a token (though a node may contain arguably more information) or each graph in PCQM4M as an image, these graph datasets are at least 10^3 to 10^4 times smaller than their NLP or CV counterparts.

In addition to the data utilized for pre-training, commonly accepted and widely adopted benchmarks, such as SuperGLUE [31] and BIG-bench [32] for NLP and ImageNet [33] for CV, have been found to be beneficial in the development of large models. These benchmarks are especially useful in assessing model quality and determining the most promising technical routes during the early stages. Although there are numerous benchmarks available for graph learning, such as Open Graph Benchmark [30] and Benchmarking GNN [34], it is likely that their scope, including factors like scale, task and domain diversity, and evaluation protocols, may not be suitable or sufficient for evaluating large graph models. Therefore, the creation of more specialized benchmarks can further facilitate the progress of large graph models.

In summary, the availability of high-quality graph data is critical to the development of large graph models, which requires more resources and efforts. Since collecting such graph data is difficult and costly, community-wide collaboration may be essential to accelerate this process.

5 Graph Models

In this section, we continue the discussion from the graph model aspect. Similar to large models in other domains, we divide our discussion into three topics: backbone architecture, pre-training, and post-processing. We also discuss LLMs as graph models, which is a recently trending direction.

5.1 Backbone Architecture

To date, Transformers [35] have been the de facto standards for NLP and CV. However, no similar consensus has been reached for the graph domain. We briefly discuss two promising deep learning architectures for graphs: graph neural networks (GNNs) and graph transformers.

GNNs are the most popular deep learning architectures for graphs [36] and have been extensively studied. Most representative GNNs adopt a message-passing paradigm, where nodes exchange messages with their neighbors to update their representations. GNNs can incorporate both structural information and semantic information such as node and edge attributes in an end-to-end manner. However, despite achieving considerable successes in many graph tasks, one key obstacle for further advancing GNNs into large models is their limited model capacity. As opposed to the scaling law in large models [16], the performance of GNNs saturates or even dramatically drops as the model size grows. Many research efforts have been devoted to explain this problem, such as over-smoothing [37] and over-squashing [38], as well as strategies to alleviate it. Nevertheless, progress has not been groundbreaking. To date, most successful GNNs only have at most millions of parameters, and further scaling to billions of parameters leads to minimum or no additional improvement.

Graph Transformer is another architecture that extends and adapts the typical Transformers for graph data [39]. In a nutshell, since classical Transformers cannot naturally process graph structures, Graph Transformer adopts various structure-encoding strategies to add graph structures to the input of

Transformers [40]. Graph Transformers evaluate the importance of each neighboring node, giving larger weights to nodes that provide more pertinent information. The self-attention mechanism empowers Graph Transformers the ability to dynamically adapt. One of the most successful graph Transformers is Graphormer [41], which ranked first in the PCQM4M molecule property prediction task of OGB Large-Scale Challenge [42] in 2021. More efforts further improve Graph Transformer from various aspects including architecture designs, efficiency, model expressiveness, etc. For example, Structure-Aware Transformer (SAT) [43] proposes a new self-attention mechanism to capture the structural similarity between nodes more effectively. AutoGT [40] proposes a unified graph transformer formulation for existing graph transformer architectures and enhances the model performance using AutoML. To improve efficiency, General, Powerful, and Scalable graph Transformer (GPS) [44] introduces a general framework with linear complexity by decoupling the local edge aggregation from the fully-connected Transformer. NAGphormer [45] also aims to address the complexity challenge of graph Transformers for large graphs by treating different hops of neighbors as a sequence of token vectors. For the expressiveness, SEG-WL test [46] introduces a graph isomorphism test algorithm, which can be used for assessing the structural discriminative power of graph Transformers. FeTA [47] analyzes the expressiveness of graph Transformers in the spectral domain and proposes to perform attention on the entire graph spectrum.

We briefly summarize the key differences between GNNs and graph Transformers, while more discussions for the relationships between transformers and GNNs can be found [48, 49, 50, 51]:

- Aggregation vs. Attention: GNNs employ message passing functions to aggregate information from neighboring nodes, whereas Graph Transformers weigh contributions from neighbors using self-attentions, potentially enhancing the flexibility for large graph models.
- Modeling structures: GNNs naturally incorporate graph structures in the message passing functions as an inductive bias, while graph Transformers adopt pre-processing strategies, such as structure-encoding, to incorporate structures.
- Depth and Over-smoothing: As aforementioned, deep GNNs may suffer from over-smoothing, leading to a decrease in their discriminative power. Graph Transformers, on the other hand, do not exhibit similar issues empirically. One plausible explanation is that Graph Transformers adaptively focus on more relevant nodes, enabling them to effectively filter and capture informative patterns.
- Scalability and Efficiency: GNNs, with their relatively simpler operations, may offer computational benefits for certain tasks. In contrast, the self-attention mechanism between node pairs in Graph Transformers can be computationally intensive, especially for large graphs. Considerable efforts have been dedicated to further enhancing the scalability and efficiency for both methods.

While both GNNs and Graph Transformers have made remarkable progress, it is not very clear which one, or some other architectures, may be best suited as the backbone for large graph models. Besides empirical evidence from trials and errors, further research into how large models work and what graph problems they may solve could bring principled advancements. It is also worth noting that most graph tasks relate to reasoning rather than perception. Therefore, the inductive bias in architecture designs usually does not come from mimicking human brains.

In our opinion, given the scale of existing graph datasets, GNNs are still a strong backbone model thanks to their strong inductive bias and expressive power. However, as the size of the training graph datasets continues to increase, graph Transformers may become more powerful through increasing the number of parameters and gradually become the prevailing approach.

5.2 Pre-training

Pre-training, as a widely adopted practice in NLP with well-known models like BERT [2] and GPT [52], involves training a model on a massive dataset before applying it for specific tasks. The primary objective is to capture general patterns or knowledge present in the data and subsequently adapt the pre-trained model to meet downstream requirements. Graph pre-training, also known as unsupervised or self-supervised graph learning, has received significant attention in recent years [53, 54]. It aims to capture the inherent structural patterns within the training graph data, analogous to how language models capture the syntax and semantics of languages. As explained in Section 2, we recognize pre-training as an essential paradigm for large graph models. Next, we provide a more detailed discussion of graph pre-training.

Compared to the straightforward yet effective masking operation used in language modeling, graph pre-training strategies are more diverse and complicated, ranging from contrastive to predictive/generative approaches. Generally, graph pre-training methods leverage the rich structural and semantic information in the graph to introduce pretext learning tasks. Through these tasks, the pre-trained model learns useful node, edge, or graph-level representations without relying on explicitly annotated labels. In contrastive pre-training methods, positive and negative graph samples are constructed through various graph data augmentation techniques, followed by optimizing contrastive objectives, such as maximizing the mutual information between positive and negative pairs. On the other hand, in generative and predictive methods, specific components of the graph data, such as node features and edges, are first hide by masking. Then, the graph model aims to reconstruct the masked portions, which serve as pseudo-labels for pre-training. For more details, we refer readers to dedicated surveys [53, 54].

We summarize the desired benefits of graph pre-training using the following “four-E" principle:

- **E**ncoding structural information: Unlike pre-training methods for other types of data, such as languages and images, which focus primarily on semantic information, graphs contain rich structural information. Pre-training large graph models essentially needs to integrate structural and semantic information from diverse graph datasets. This also highlights the unique challenges and opportunities of graph pre-training.
- **E**asing data sparsity and label scarcity: Large graph models, with their substantial model capacity, are prone to overfitting when confronted with specific tasks that have limited labeled data. Pre-training on a wide range of graph datasets and tasks can act as a regularizing mechanism, preventing the model from overfitting to a specific task and improving generalization performance.
- **E**xpanding applicability domains: One of the hallmarks of pre-training is the ability to transfer learned knowledge across various domains. By pre-training large graph models on diverse graph datasets, they should be able to capture a wide range of structural patterns, which can then be applied, adapted, or fine-tuned to graph data in similar domains, maximizing the model's utility.
- **E**nhancing robustness and generalization. Pre-training methods can expose large graph models to diverse graphs with distinct characteristics, including varying sizes, structures, and complexities. This exposure can potentially lead to more robust models that are less sensitive to adversarial perturbations [55]. Moreover, models trained in this manner are more likely to generalize well to unseen graph data or novel graph tasks.

In summary, graph pre-training is not merely a beneficial or supplementary step, but a pivotal and necessary paradigm for large graph models.

5.3 Post-processing

After obtaining a substantial amount of knowledge through pre-training, LLMs still require post-processing to enhance their adaptability to downstream tasks. Representative post-processing techniques include prompting [56], parameter-efficient fine-tuning [57], reinforcement learning with human feedbacks [58], and model compression [59]. For graphs, some recent efforts have also been devoted to study post-processing techniques for pre-trained models.

Prompting originally refers to methods that provide specific instructions to language models for generating desired contents for downstream tasks. Recently, constructing prompts with an in-context learning template demonstrates great effectiveness in LLMs [60]. Language prompts usually contain a task description and a few examples to illustrate the downstream tasks. Graph prompts, which mimic natural language prompts to enhance downstream task performance with limited labels and enable interaction with the model to extract valuable knowledge, have been extensively studied [61]. One significant challenge for graph prompts is the unification of diverse graph tasks, spanning from node-level and link-level to graph-level tasks. In contrast, tasks in natural language can be easily unified as language modeling under specific constraints. To tackle this challenge, GPPT [62] unifies graph tasks into edge prediction, considering that a typical node classification task can be reformulated as the link prediction task between the structure-token and the task-token. Each structure-token represents a node in the graph data, and each task-token corresponds to a class. GraphPrompt [61] further extends the idea and unifies link prediction, node classification, and graph classification as subgraph similarity calculation by describing node and graph classes as prototypical subgraphs. Similarly, ProG [63] reformulates node and edge-level tasks as graph-level tasks and further proposes multi-task prompting by realizing prompting as a learnable token that is directly added to the node feature, mirroring the prefix phrase prompting technique in NLP. ProG also employs meta learning to learn prompting for different tasks. Other graph prompts such as PRODIGY [64], GPF [65], Gare [66], SGL-PT [67], DeepGPT [68], and G-Prompt [69] follow similar principles.

Parameter-efficient fine-tuning refers to techniques where only a small portions of model parameters are optimized, while the rest is kept fixed. Besides reducing computational costs, it also helps to enable the model to adapt to new tasks without forgetting the knowledge obtained in pre-training, preserving the general capabilities of the model while allowing for task-specific adaptation. Graph parameter-efficient fine-tuning has also recently begun to received attention. For example, AdapterGNN [70] and G-Adapter [71] both investigate adapter-based fine-tuning techniques for graph models, aiming to reduce the number of tuneable parameters while preserving comparable accuracy. Specifically, AdapterGNN tunes GNNs by incorporating two adapters, one inserted before and another one after the message passing process. On the other hand, G-Adapter focuses on graph transformers and introduces a message passing process within the adapter to better utilize graph structural information. S2PGNN [72] further proposes to search for architecture modifications to improve the adaptivity of the fine-tuning stage.

Model compression aims to reduce the memory and computational demands of models through various techniques, including knowledge distillation, pruning, and quantization, which are particularly valuable when deploying large models in resource-constrained environments. Here, we focus on quantization, which has gained popularity and proven effectiveness in LLMs [3], and refer readers to dedicated surveys for other methods [73, 74, 75]. Quantization entails reducing the precision of numerical values used by the model while preserving model performance to the greatest extent possible. In the case of large models, post-training quantization (PTQ) is particularly preferred, as it does not require retraining. PTQ in graph learning has also been explored in SGQuant [76], which proposes a multi-granularity quantization technique that operates at various levels, including graph topology, layers, and components within a layer. Other methods such as Degree-Quant [77], BiFeat [78], Tango [79], VQGraph [80], A^2Q [81], and AdaQP [82] adopt a quantization-aware training scheme, which are inspiring but cannot be used standalone during the post-processing stage.

In summary, the success of post-processing techniques shown in LLMs has sparked interest in similar research in the graph domain. However, due to the unavailability of large graph models at present, the assessment of these methods is limited to relatively small models. Therefore, it is crucial to further verify their effectiveness when applied to large graph models, and more research challenges and opportunities may arise.

5.4 LLMs as Graph Models

Recent research has also explored the potential of directly utilizing LLMs for solving graph tasks. The essential idea is to transform graph data, including both graph structures and features, as well as graph tasks, into natural language representations, thereby treating graph problems as regular NLP problems. In the following discussion, we provide a brief overview of these advancements. More detailed discussions are provided at Appendix B.

NLGraph [83] conducts a systematic evaluation of LLMs, such as GPT-3 and GPT-4, on eight graph reasoning tasks in natural language, spanning varying levels of complexity, including connectivity, shortest path, maximum flow, simulating GNNs, etc. It empirically finds that LLMs show preliminary graph reasoning abilities, but struggle with more complex graph problems, potentially because they solely capture spurious correlations within the problem settings. Meanwhile, GPT4Graph [84] also conducts extensive experiments to evaluate the graph understanding capabilities of LLMs across ten distinct tasks, such as graph size and degree detection, neighbor and attribute retrieval, etc. It reveals the limitations of LLMs in graph reasoning and emphasizes the necessity of enhancing their structural understanding capabilities. LLMtoGraph [85] also tests GPT-3.5 and GPT-4 for various graph tasks and makes some interesting observations.

More recently, Graph-LLM [86] systematically investigates the utilization of LLMs in text-attributed graph through two strategies: LLMs-as-Enhancers, where LLMs enhance the representations of text attributes of nodes before passing them to GNNs, and LLMs-as-Predictors, where LLMs are directly employed as predictors. Comprehensive studies have been conducted on these two pipelines across various settings, and the empirical results provide valuable insights into further leveraging LLMs for graph machine learning. InstructGLM [87] further introduces scalable prompts designed to describe the graph structures and features for LLM instruction tuning, which enables tuned LLMs to perform various graph tasks during the inference stage in a generative manner. Experiments conducted on GNN benchmarks empirically show the strong potential of adopting LLMs for graph machine learning.

Although still in their early stages, these works highlight that LLMs also represent a promising avenue for developing large graph models, which is worthy further exploration and investigation.

5.5 Summary

To summarize, substantial research efforts have been devoted to studying various aspects of graph models. However, there is currently no clear framework for effectively integrating these techniques into large graph models. Consequently, more efforts are required to compare existing methods and develop advanced models. In this endeavor, automated graph machine learning techniques [88], such as graph neural architecture search, can be valuable in reducing human effort and accelerating the trial-and-error process.

6 Applications

Instead of attempting to overwhelmingly handle various graph domains and tasks, it may be more effective to focus on specific graph-related vertical fields by leveraging domain knowledge and domain-specific datasets. In this section, we highlight several graph application scenarios that can significantly benefit from large graph models.

6.1 Recommendation System

Graph data naturally exists in recommendation systems. For example, the interaction between users and items can be modeled as a bipartite graph or more complex heterogeneous graphs that include clicks, buys, reviews, and more. Currently, LLMs for recommendation systems focus on modeling semantic information [89], while explicitly utilizing the structural information of graphs has the potential to yield better results [90]. A potential challenge is that graphs in recommendation system are usually multi-modal [91], covering text, images, interactions, etc. Since large models for multi-modal data are not yet mature, significant efforts are needed to develop truly effective large graph models for recommendation systems.

6.2 Knowledge Graph

Knowledge graphs are widely adopted to store and utilize ubiquitous knowledge in human society. LLMs have been used for various knowledge graph tasks [92, 87], including construction, completion, and question answering. Despite their achievements, most of these methods focus primarily on the textual information, leaving the structural and relational information of knowledge graphs under-explored. Large graph models, potentially combined with existing LLMs, can greatly complement the status quo and further promote research and application of knowledge graphs.

6.3 Molecules

Graphs are natural representations for molecules, where nodes represent atoms and edges indicate bonds. Building effective graph models for molecules can advance various applications, including molecular property prediction and molecular dynamics simulations, ultimately benefiting drug discovery. Currently, some variants of LLMs are applied to molecules [93, 94] by first transforming molecules into strings using SMILES [95], which allows molecules to be represented and generated as regular texts. Nevertheless, graphs serve as a more natural way to represent the structural information of molecules with numerous modeling advantages [96]. Meanwhile, a great number of graph-based pre-training techniques have also been developed for molecules [97], including multi-modal strategies [98]. Besides, molecule data is relatively easier to collect, e.g., ZINC20 [99] contains millions of purchasable compounds. Therefore, we believe graph-based or graph-enhanced large models for molecule modeling can soon to be expected.

6.4 Finance

Graph machine learning has proven to be beneficial for multiple financial tasks such as stock movement prediction and loan risk prediction [100]. Moreover, the large abundance of financial data makes it possible to construct domain-specific large models, exemplified by BloombergGPT [101]. By combining the strengths of both worlds, the application of large graph models in the field of finance holds great promise. A potential challenge lies in the sensitive and private nature of most financial data, making industries reluctant to release related models and data to the public. Efforts are required to promote open-source initiatives and democratization [102, 103] to fully unleash the potential of large graph models in the finance area.

6.5 Code and Program

Thanks to the large amount of code data available on repository hosting platforms such as GitHub, LLMs show remarkable ability in understanding and generating codes and programs. Notable examples include CodeX [104], AlphaCode [105], and GPT-4 [29], which have exerted a significant impact on the programming landscape, potentially even reshaping it. In addition to treating codes and programs as textual data, graphs offer a natural means to represent the structural aspects of codes. For example, abstract syntax trees, including control flow graph, data flow graph, etc., effectively capture the syntactic structure of source codes [106]. Studies have demonstrated that the integration of graphs can further enhance the performance of LLMs by providing complementary information [107]. Therefore, large graph models hold valuable potential for a wide range of code and program-related tasks, including code completion and generation, code search, code review, program analysis and testing, among others.

6.6 Urban Computing and Transportation

Graph data is pervasive in the domains of urban computing and transportation, such as road networks. Therefore, graph machine learning can benefit many applications, including traffic forecasting, various urban planning and management tasks, crime prediction, and epidemic control [108, 109]. Moreover, large-scale urban data naturally exists, such as mobility data collected from GPS and diverse sensors. Currently, some LLM-based large models have been explored for urban computing and transportation, such as TransGPT [110]. Nevertheless, their focus has primarily revolved around natural language related applications, leaving developing large graph models for broader and more comprehensive utilization still an open opportunity. One major technical challenge in the process lies in that graph data in urban and transportation contexts is dynamic in nature, containing complicated spatial-temporal patterns. Thus, a large graph model needs to effectively capture both structural and temporal information to achieve satisfactory performance.

6.7 Beyond

The application scenarios we have outlined above are by no means exhaustive. Considering that graph machine learning has been widely adopted across diverse domains ranging from industrial applications, such as fault diagnosis [111], IoT [112], power systems [113], and time-series analysis [114], to AI for science [115], such as physics [116, 117], combinatorial optimization [118], material science [119], and neural science [120], exploring the usage of large graph models holds extremely rich potentials.

7 Conclusion

In summary, large graph models can potentially revolutionize the field of graph machine learning, but they also give rise to a multitude of challenges, ranging from the representation basis, graph data, graph models, and applications. Meanwhile, promising endeavors are being undertaken to tackle these challenges, creating exciting opportunities for both researchers and practitioners. We hope that our perspective will inspire continued efforts and advancements for large graph models.

Acknowledgement

This work was supported by the National Key Research and Development Program of China No. 2020AAA0106300, National Natural Science Foundation of China (No. 62222209, 62250008, 62102222, 62206149), Beijing National Research Center for Information Science and Technology under Grant No. BNR2023RC01003, BNR2023TD03006, China National Postdoctoral Program for Innovative Talents No. BX20220185, China Postdoctoral Science Foundation No. 2022M711813, and Beijing Key Lab of Networked Multimedia. All opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies. Xin Wang and Wenwu Zhu are corresponding authors.

A An Illustration of Desired Characteristics

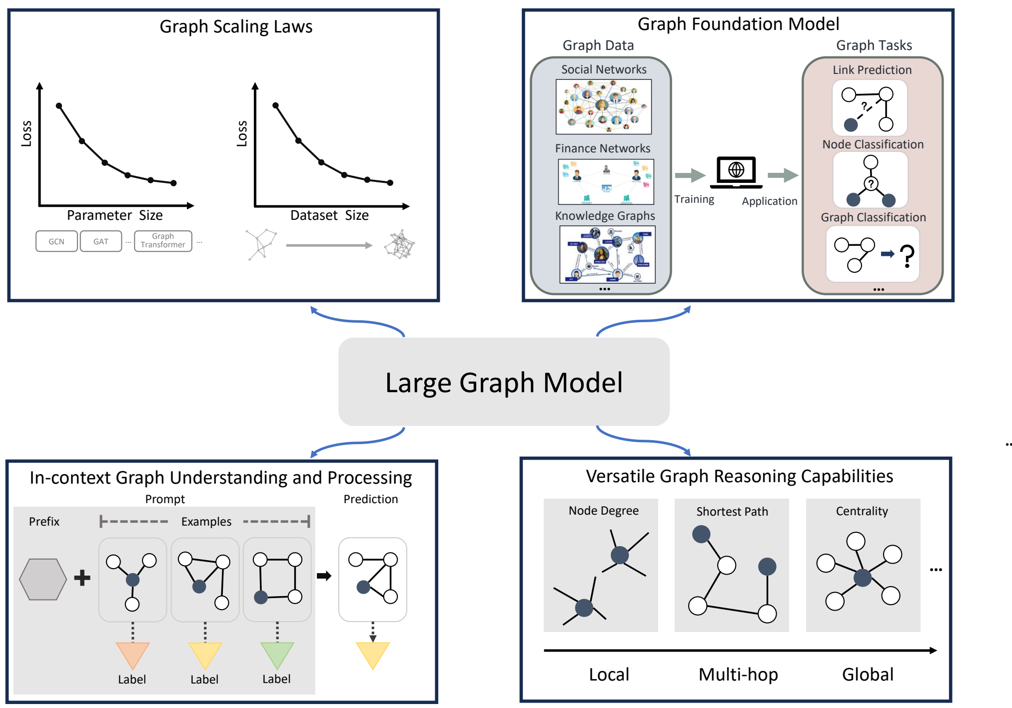

Figure 1: An illustration of desired characteristics of a large graph model.

B Detailed Discussions for LLMs as Graph Models

In this section, we summarize and compare different models related to LLMs as graph models. The overall summarization is shown in Table 1. Specifically, we category the key features into three groups: model architectures, modeling Graph structure for LLMs, and graph data.

For model architectures, we summarize the following designs:

- Final Predictor: whether the model utilizes GNNs or LLMs to get the final prediction.
- LLMs: which LLMs are utilized in the model. Typical examples include GPT-3 [1], GPT-4 [29], Llama 2 [121], etc.
- Need Fine-tuning: whether the model needs to be fine-tuned. Note that if the model does not necessarily require fine-tuning, but could be fine-tuned to further improve the performance, we mark it as no. Note that close-sourced LLMs such as GPT-3 and GPT-4 are not tunable.

- Tunable Components: which parts of the model can be fine-tuned, such as GNNs, graph Transformers, and LLMs.
- Receptive Field: how many hops of neighbors can be perceived when making predictions. K-hop indicates the receptive field is determined by the architectures, e.g., the number of layers in GNNs.

One key challenge of using LLMs as graph models is to model graph structures and inject them into LLMs. As this is usually achieved through prompts, we make the following summarization:

- Prompt Type: whether the model uses textual prompts (i.e., descriptions in texts) or neural prompts (e.g., through hidden layers in neural networks).
- Prompt Details: the details of the prompt. Common textual prompts include adjacency lists and neighborhood descriptions, and typical neural prompts include GNNs and graph Transformers.
- Advanced graph-specific prompt: the types of advanced graph-specific prompts, if they are proposed in the paper.

Lastly, we summarize the graph data used in the experiments. Note that we focus on the experiments conducted in the original papers, but extensions are possible, e.g., handling larger graphs by using more computational resources or applying the model to other tasks through minor modifications.

- Dataset Type: what type of graphs are utilized in the experiments, including synthetic graphs, TAGs, knowledge graphs (KGs), and general graphs.
- Tasks: what type of tasks are considered in the experiments, including algorithmic tasks (various from degree counting to finding shortest paths, etc.), node classification, link prediction, question answering (QA), etc.
- #Nodes: the approximate number of nodes handled by the model. If sampling is applied, we only count the sampled nodes.
- Node Feature: whether and what type of node features can be utilized in the model, including no attributes, text attributes, and general attributes.

C Graph Data

In this section, we summarize some principles that are helpful while collecting more graph data.

- Domain diversity: To enable large graph models handle different graph applications, it is crucial to expose the model to different domains of interests, so that large graph model can be adopted across various fields.
- Type diversity: Graphs have rich types, including homogeneous and heterogeneous, homophily and heterophily, static and dynamic, directed and undirected, weighted and unweighted, signed and unsigned, etc. The diversity of graph type is also important to empower the large graph model handle diverse downstream graphs.
- Statistics diversity: Graphs also have varying statistics, e.g., size, density, degree distribution, etc. Such diversity should be considered to ensure the effectiveness of large graph model.
- Task diversity: Graph tasks are also distinct, ranging from node-level, edge-level to graph-level, and from discriminative tasks such as classification and prediction to generative tasks such as graph generation. Increasing the task diversity in pre-training or post-processing phase can help developing effective large graph models.
- Modality diversity: Graphs can combine different modalities of data, such as text, images, and tabular data, which can enrich the utility of the large graph model.

Table 1: A summarization of different models related to LLMs as graph models.

Method	Architecture					Modeling Graph Structure for LLMs			Graph Data			
	Final Predictor	LLMs	Need Finetune	Tunable Components	Receptive Field	Prompt Type	Prompt Details	Advanced Graph-specific Prompt	Dataset Type	Tasks	# Nodes	Node Feature
NLGraph [83]	LLM	GPT-3/4	✗	-	Full	Text	Edge list	Build-a-Graph Prompting Algorithmic Prompting	Synthetic	Algorithm	10^2	No
GPT4Graph [84]	LLM	GPT-3	✗	-	2-hop	Text	Graph modelling language graph markup language	-	Synthetic, KG	Algorithm, QA	10^2	Text
Graph-LLM(predictor) [86]	LLM	GPT-3	✗	-	2-hop	Text	Neighbor description	Neighbor Summary	TAG	Node classification	10^2	Text
Graph-LLM(enhancer) [86]	GNN	GPT-3, E5, etc.	✓	GNN/LLM	K-hop		-	-	TAG	Node classification	10^6	General
LLMtoGraph [85]	LLM	GPT-3/4, Vicuna, etc.	✗	-	Full	Text	Edge list	-	Synthetic	Algorithm	10^2	No
InstructGLM [87]	LLM	T5, LLama-2	✓	LLM	2-hop	Text	Neighbor description	-	TAG	Node classification	10^5	Text
LLM-Structured-Data [122]	LLM	GPT-3	✗	-	2-hop	Text	Neighbor description	Attention extraction/prediction	TAG	Node classification	10^3	Text
OneForAll [123]	GNN	GPT-3,Bert	✓	GNN	K-hop	Text	-	-	TAG,KG	Node classification Link Prediction Graph Classification	10^5	Text
GraphText [124]	LLM	GPT-3, Llama-2	✗	LLM	2-hop	Text	Graph syntax tree + GNN	Graph Syntax Tree	General	Node classification	10^3	General
GraphQA [125]	LLM	PaLM-2	✗	LLM	Full	Text	Task-specific relation encoding	-	Synthetic	Algorithm	10^2	No
Hu et al. [126]	LLM	GPT-3/4	✗	-	1-hop	Text	Neighbor description	-	TAG	Node classification Link Prediction Graph Classification	10^5	Text
GraphLLM [127]	LLM	LLama-2	✓	GT	Full	Neural	Graph Transformer	-	Synthetic	Algorithm	10^2	Text
LLM-GNN [128]	GNN	GPT-3	✓	GNN	K-hop		-	-	TAG	Node classification	10^6	General
ENG [129]	GNN	GPT-3, Llama-2	✓	GNN	K-hop	Text	Neighbor description	-	General	Node classification	10^5	General
Wenkel et al. [130]	LLM	GPT-2/3	✗	LLM	Full	Text	Edge list	-	Synthetic Molecules	Graph classification	10^2	Text
GraphGPT [131]	LLM	Llama-2, Vicuna	✓	GNN	2-hop	Neural	GNN + Instruction tuning	-	TAG	Node classification Link Prediction Graph Matching	10^5	General
LLM4DyG [132]	LLM	GPT-3, Llama-2, etc.	✗	-	Full	Text	Temporal edge list	Disentangled Spatial-Temporal Thoughts	Synthetic	Dynamic Algorithm	10^2	No
DGTL [133]	LLM	LLama-2	✓	GNN	K-hop	Neural	Disentagled GNN	-	TAG	Node classification	10^4	Text
Stechly et al. [134]	LLM	GPT-4	✗	-	Full	Text	Edge list	Iterative prompting	Synthetic	Graph Coloring	10^2	No
SimTeG [27]	GNN	E5,Roberta	✓	GNN	K-hop		-	-	TAG	Node classification Link Prediction	10^6	Text
GNP [135]	LLM	T5	✓	GNN/LLM	K-hop	Neural	GNN + subgraph retrival	-	KG	QA	10^4	Text
RoG [136]	LLM	GPT-3/ Llama-2	✓	LLM	L-hop	Text	Retriveal reasoning	-	KG	QA	10^4	Text
TAPE [26]	GNN	GPT-3	✓	GNN/LLM	K-hop		-	-	TAG	Node classification	10^5	Text

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901, 2020.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019.

[3] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763, 2021.

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[7] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph. *arXiv preprint arXiv:2307.07697*, 2023.

[8] Yao Yao, Zuchao Li, and Hai Zhao. Beyond chain-of-thought, effective graph-of-thought reasoning in large language models. *arXiv preprint arXiv:2305.16582*, 2023.

[9] Fanglong Yao, Changyuan Tian, Jintao Liu, Zequn Zhang, Qing Liu, Li Jin, Shuchao Li, Xiaoyu Li, and Xian Sun. Thinking like an expert: Multimodal hypergraph-of-thought (hot) reasoning to boost foundation modals. *arXiv preprint arXiv:2308.06207*, 2023.

[10] Lang Cao. Enhancing reasoning capabilities of large language models: A graph-based verification approach. *arXiv preprint arXiv:2308.09267*, 2023.

[11] Bin Lei, pei Hung Lin, Chunhua Liao, and Caiwen Ding. Boosting logical reasoning in large language models through a new framework: The graph of thought. *arXiv preprint arXiv:2308.08614*, 2023.

[12] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*, 2023.

[13] Yilin Wen, Zifeng Wang, and Jimeng Sun. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. *arXiv preprint arXiv:2308.09729*, 2023.

[14] Jiawei Zhang. Graph-toolformer: To empower llms with graph reasoning ability via prompt augmented by chatgpt. *arXiv preprint arXiv:2304.11116*, 2023.

[15] Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. Structgpt: A general framework for large language model to reason over structured data. *arXiv preprint arXiv:2305.09645*, 2023.

[16] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[17] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

[18] Xiaojie Guo and Liang Zhao. A systematic survey on deep generative models for graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5370–5390, 2023.

[19] Chuxu Zhang, Kaize Ding, Jundong Li, Xiangliang Zhang, Yanfang Ye, Nitesh V Chawla, and Huan Liu. Few-shot learning on graphs. In *31st International Joint Conference on Artificial Intelligence, IJCAI 2022*, pages 5662–5669, 2022.

[20] Mingxuan Ju, Tong Zhao, Qianlong Wen, Wenhao Yu, Neil Shah, Yanfang Ye, and Chuxu Zhang. Multi-task self-supervised graph neural networks enable stronger task generalization. In *The Eleventh International Conference on Learning Representations*, 2022.

[21] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Out-of-distribution generalization on graphs: A survey. *arXiv preprint arXiv:2202.07987*, 2022.

[22] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[23] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):5782–5799, 2022.

[24] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2303.18223*, 2022.

[25] Mark Newman. *Networks*. Oxford university press, 2018.

[26] Xiaoxin He, Xavier Bresson, Thomas Laurent, and Bryan Hooi. Explanations as features: Llm-based features for text-attributed graphs. *arXiv preprint arXiv:2305.19523*, 2023.

[27] Keyu Duan, Qian Liu, Tat-Seng Chua, Shuicheng Yan, Wei Tsang Ooi, Qizhe Xie, and Junxian He. Simteg: A frustratingly simple approach improves textual graph learning. *arXiv preprint arXiv:2308.02565*, 2023.

[28] Petar Veličković and Charles Blundell. Neural algorithmic reasoning. *Patterns*, 2(7):100273, 2021.

[29] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[30] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[31] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

[32] BIG bench authors. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

[33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009.

[34] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[36] Ziwei Zhang, Peng Cui, and Wenwu Zhu. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):249–270, 2020.

[37] T Konstantin Rusch, Michael Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks. *SAM Research Report*, 2023, 2023.

[38] Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022.

[39] Erxue Min, Runfa Chen, Yatao Bian, Tingyang Xu, Kangfei Zhao, Wenbing Huang, Peilin Zhao, Junzhou Huang, Sophia Ananiadou, and Yu Rong. Transformer for graphs: An overview from architecture perspective. *arXiv preprint arXiv:2202.08455*, 2022.

[40] Zizhao Zhang, Xin Wang, Chaoyu Guan, Ziwei Zhang, Haoyang Li, and Wenwu Zhu. Autogt: Automated graph transformer architecture search. In *The Eleventh International Conference on Learning Representations*, 2023.

[41] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

[42] Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. Ogb-lsc: A large-scale challenge for machine learning on graphs. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[43] Dexiong Chen, Leslie O'Bray, and Karsten Borgwardt. Structure-aware transformer for graph representation learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 3469–3489, 2022.

[44] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

[45] Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. NAGphormer: A tokenized graph transformer for node classification in large graphs. In *The Eleventh International Conference on Learning Representations*, 2023.

[46] Wenhao Zhu, Tianyu Wen, Guojie Song, Liang Wang, and Bo Zheng. On structural expressive power of graph transformers. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 3628–3637, 2023.

[47] Anson Bastos, Abhishek Nadgeri, Kuldeep Singh, Hiroki Kanezashi, Toyotaro Suzumura, and Isaiah Onando Mulang. How expressive are transformers in spectral domain for graphs? *Transactions on Machine Learning Research*, 2022.

[48] Chaitanya Joshi. Transformers are graph neural networks. *The Gradient*, 2020.

[49] Petar Veličković. Everything is connected: Graph neural networks. *Current Opinion in Structural Biology*, 79:102538, 2023.

[50] Luis Müller, Mikhail Galkin, Christopher Morris, and Ladislav Rampášek. Attending to graph transformers. *arXiv preprint arXiv:2302.04181*, 2023.

[51] Jinwoo Kim, Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, and Seunghoon Hong. Pure transformers are powerful graph learners. *Advances in Neural Information Processing Systems*, 35:14582–14595, 2022.

[52] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[53] Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and S Yu Philip. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):5879–5900, 2022.

[54] Lirong Wu, Haitao Lin, Cheng Tan, Zhangyang Gao, and Stan Z Li. Self-supervised learning on graphs: Contrastive, generative, or predictive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[55] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International conference on machine learning*, pages 2712–2721, 2019.

[56] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[57] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

[58] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[59] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *arXiv preprint arXiv:2308.07633*, 2023.

[60] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[61] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM Web Conference 2023*, pages 417–428, 2023.

[62] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1717–1727, 2022.

[63] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. All in one: Multi-task prompting for graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining (KDD'23)*, 2023.

[64] Qian Huang, Hongyu Ren, Peng Chen, Gregor Kržmanc, Daniel Zeng, Percy Liang, and Jure Leskovec. Prodigy: Enabling in-context learning over graphs. *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[65] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal prompt tuning for graph neural networks. *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[66] Yongcheng Jing, Chongbin Yuan, Li Ju, Yiding Yang, Xinchao Wang, and Dacheng Tao. Deep graph reprogramming. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24345–24354, 2023.

[67] Yun Zhu, Jianhao Guo, and Siliang Tang. Sgl-pt: A strong graph learner with graph prompt tuning. *arXiv preprint arXiv:2302.12449*, 2023.

[68] Reza Shirkavand and Heng Huang. Deep prompt tuning for graph transformers. *arXiv preprint arXiv:2309.10131*, 2023.

[69] Xuanwen Huang, Kaiqiao Han, Dezheng Bao, Quanjin Tao, Zhisheng Zhang, Yang Yang, and Qi Zhu. Prompt-based node feature extractor for few-shot learning on text-attributed graphs. *arXiv preprint arXiv:2309.02848*, 2023.

[70] Shengrui Li, Xueting Han, and Jing Bai. Adaptergnn: Efficient delta tuning improves generalization ability in graph neural networks. *arXiv preprint arXiv:2304.09595*, 2023.

[71] Anchun Gui, Jinqiang Ye, and Han Xiao. G-adapter: Towards structure-aware parameter-efficient transfer learning for graph transformer networks. *arXiv preprint arXiv:2305.10329*, 2023.

[72] Zhili Wang, Shimin Di, Lei Chen, and Xiaofang Zhou. Search to fine-tune pre-trained graph neural networks for graph-level tasks. *arXiv preprint:2308.06960*, 2023.

[73] Yijun Tian, Shichao Pei, Xiangliang Zhang, Chuxu Zhang, and Nitesh V. Chawla. Knowledge distillation on graphs: A survey. *arXiv preprint arXiv:2302.00219*, 2023.

[74] Shichang Zhang, Atefeh Sohrabizadeh, Cheng Wan, Zijie Huang, Ziniu Hu, Yewen Wang, Yingyan, Lin, Jason Cong, and Yizhou Sun. A survey on graph neural network acceleration: Algorithms, systems, and customized hardware. *arXiv preprint arXiv:2306.14052*, 2023.

[75] Sergi Abadal, Akshay Jain, Robert Guirado, Jorge López-Alonso, and Eduard Alarcón. Computing graph neural networks: A survey from algorithms to accelerators. *ACM Computing Surveys*, 54(9), 2021.

[76] Boyuan Feng, Yuke Wang, Xu Li, Shu Yang, Xueqiao Peng, and Yufei Ding. Sgquant: Squeezing the last bit on graph neural networks with specialized quantization. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence*, pages 1044–1052, 2020.

[77] Shyam Anil Tailor, Javier Fernandez-Marques, and Nicholas Donald Lane. Degree-quant: Quantization-aware training for graph neural networks. In *International Conference on Learning Representations*, 2020.

[78] Yuxin Ma, Ping Gong, Jun Yi, Zhewei Yao, Cheng Li, Yuxiong He, and Feng Yan. Bifeat: Supercharge gnn training via graph feature quantization. *arXiv preprint arXiv:2207.14696*, 2022.

[79] Shiyang Chen, Da Zheng, Caiwen Ding, Chengying Huan, Yuede Ji, and Hang Liu. Tango: rethinking quantization for graph neural network training on gpus. *arXiv preprint arXiv:2308.00890*, 2023.

[80] Ling Yang, Ye Tian, Minkai Xu, Zhongyi Liu, Shenda Hong, Wei Qu, Wentao Zhang, Bin Cui, Muhan Zhang, and Jure Leskovec. Vqgraph: Graph vector-quantization for bridging gnns and mlps. *arXiv preprint arXiv:2308.02117*, 2023.

[81] Zeyu Zhu, Fanrong Li, Zitao Mo, Qinghao Hu, Gang Li, Zejian Liu, Xiaoyao Liang, and Jian Cheng. a^2q: Aggregation-aware quantization for graph neural networks. In *The Eleventh International Conference on Learning Representations*, 2022.

[82] Borui Wan, Juntao Zhao, and Chuan Wu. Adaptive message quantization and parallelization for distributed full-graph gnn training. *Proceedings of Machine Learning and Systems*, 5, 2023.

[83] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can language models solve graph problems in natural language? *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[84] Jiayan Guo, Lun Du, and Hengyu Liu. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*, 2023.

[85] Chang Liu and Bo Wu. Evaluating large language models on graphs: Performance insights and comparative analysis. *arXiv preprint arXiv:2308.11224*, 2023.

[86] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. Exploring the potential of large language models (llms) in learning on graphs. *arXiv preprint arXiv:2307.03393*, 2023.

[87] Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. Natural language is all a graph needs. *arXiv preprint arXiv:2308.07134*, 2023.

[88] Ziwei Zhang, Xin Wang, and Wenwu Zhu. Automated machine learning on graphs: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2021.

[89] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. A survey on large language models for recommendation. *arXiv preprint arXiv:2305.19860*, 2023.

[90] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37, 2022.

[91] Zhulin Tao, Yinwei Wei, Xiang Wang, Xiangnan He, Xianglin Huang, and Tat-Seng Chua. Mgat: Multimodal graph attention network for recommendation. *Information Processing & Management*, 57(5):102277, 2020.

[92] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *arXiv preprint arXiv:2306.08302*, 2023.

[93] Viraj Bagal, Rishal Aggarwal, PK Vinod, and U Deva Priyakumar. Molgpt: molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling*, 62(9):2064–2076, 2021.

[94] Chen Qian, Huayi Tang, Zhirui Yang, Hong Liang, and Yong Liu. Can large language models empower molecular property prediction? *arXiv preprint arXiv:2307.07443*, 2023.

[95] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[96] Oliver Wieder, Stefan Kohlbacher, Mélaine Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37:1–12, 2020.

[97] Jun Xia, Yanqiao Zhu, Yuanqi Du, Yue Liu, and Stan Z Li. A systematic survey of molecular pre-trained models. *arXiv preprint arXiv:2210.16484*, 2022.

[98] Pengfei Liu, Yiming Ren, and Zhixiang Ren. Git-mol: A multi-modal large language model for molecular science with graph, image, and text. *arXiv preprint arXiv:2308.06911*, 2023.

[99] John J Irwin, Khanh G Tang, Jennifer Young, Chinzorig Dandarchuluun, Benjamin R Wong, Munkhzul Khurelbaatar, Yurii S Moroz, John Mayfield, and Roger A Sayle. Zinc20—a free ultralarge-scale chemical database for ligand discovery. *Journal of chemical information and modeling*, 60(12):6065–6073, 2020.

[100] Jianian Wang, Sheng Zhang, Yanghua Xiao, and Rui Song. A review on graph neural network methods in financial applications. *arXiv preprint arXiv:2111.15367*, 2022.

[101] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

[102] Xiao-Yang Liu, Guoxuan Wang, and Daochen Zha. Fingpt: Democratizing internet-scale data for financial large language models. *arXiv preprint arXiv:2307.10485*, 2023.

[103] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*, 2023.

[104] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[105] Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

[106] Miltiadis Allamanis, Marc Brockschmidt, and Mahmoud Khademi. Learning to represent programs with graphs. In *International Conference on Learning Representations*, 2018.

[107] Daya Guo, Shuo Ren, Shuai Lu, Zhangyin Feng, Duyu Tang, Shujie LIU, Long Zhou, Nan Duan, Alexey Svyatkovskiy, Shengyu Fu, Michele Tufano, Shao Kun Deng, Colin Clement, Dawn Drain, Neel Sundaresan, Jian Yin, Daxin Jiang, and Ming Zhou. Graphcode{bert}: Pre-training code representations with data flow. In *International Conference on Learning Representations*, 2021.

[108] Saeed Rahmani, Asiye Baghbani, Nizar Bouguila, and Zachary Patterson. Graph neural networks for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 24(8):8846–8885, 2023.

[109] Guangyin Jin, Yuxuan Liang, Yuchen Fang, Jincai Huang, Junbo Zhang, and Yu Zheng. Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *arXiv preprint arXiv:2303.14483*, 2023.

[110] Duomo. Transgpt. https://github.com/DUOMO/TransGPT, 2023.

[111] Zhiwen Chen, Jiamin Xu, Cesare Alippi, Steven X Ding, Yuri Shardt, Tao Peng, and Chunhua Yang. Graph neural network-based fault diagnosis: a review. *arXiv preprint arXiv:2111.08185*, 2021.

[112] Guimin Dong, Mingyue Tang, Zhiyuan Wang, Jiechao Gao, Sikun Guo, Lihua Cai, Robert Gutierrez, Bradford Campbel, Laura E Barnes, and Mehdi Boukhechba. Graph neural networks in iot: a survey. *ACM Transactions on Sensor Networks*, 19(2):1–50, 2023.

[113] Wenlong Liao, Birgitte Bak-Jensen, Jayakrishnan Radhakrishna Pillai, Yuelong Wang, and Yusen Wang. A review of graph neural networks and their applications in power systems. *Journal of Modern Power Systems and Clean Energy*, 10(2):345–360, 2021.

[114] Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I Webb, Irwin King, and Shirui Pan. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *arXiv preprint arXiv:2307.03759*, 2023.

[115] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

[116] Jonathan Shlomi, Peter Battaglia, and Jean-Roch Vlimant. Graph neural networks in particle physics. *Machine Learning: Science and Technology*, 2(2):021001, 2020.

[117] Gage DeZoort, Peter W Battaglia, Catherine Biscarat, and Jean-Roch Vlimant. Graph neural networks at the large hadron collider. *Nature Reviews Physics*, pages 1–23, 2023.

[118] Quentin Cappart, Didier Chételat, Elias B Khalil, Andrea Lodi, Christopher Morris, and Petar Velickovic. Combinatorial optimization and reasoning with graph neural networks. *Journal of Machine Learning Research*, 24:130–1, 2023.

[119] Patrick Reiser, Marlen Neubert, André Eberhard, Luca Torresi, Chen Zhou, Chen Shao, Houssam Metni, Clint van Hoesel, Henrik Schopmans, Timo Sommer, et al. Graph neural networks for materials science and chemistry. *Communications Materials*, 3(1):93, 2022.

[120] Alaa Bessadok, Mohamed Ali Mahjoub, and Islem Rekik. Graph neural networks in network neuroscience. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5833–5848, 2022.

[121] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[122] Jin Huang, Xingjian Zhang, Qiaozhu Mei, and Jiaqi Ma. Can llms effectively leverage graph structural information: When and why. *arXiv preprint arXiv:2309.16595*, 2023.

[123] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. *arXiv preprint arXiv:2310.00149*, 2023.

[124] Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*, 2023.

[125] Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560*, 2023.

[126] Yuntong Hu, Zheng Zhang, and Liang Zhao. Beyond text: A deep dive into large language models' ability on understanding graph data. *arXiv preprint arXiv:2310.04944*, 2023.

[127] Ziwei Chai, Tianjie Zhang, Liang Wu, Kaiqiao Han, Xiaohai Hu, Xuanwen Huang, and Yang Yang. Graphllm: Boosting graph reasoning ability of large language model. *arXiv preprint arXiv:2310.05845*, 2023.

[128] Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. Label-free node classification on graphs with large language models (llms). *arXiv preprint arXiv:2310.04668*, 2023.

[129] Jianxiang Yu, Yuxiang Ren, Chenghua Gong, Jiaqi Tan, Xiang Li, and Xuecang Zhang. Empower text-attributed graphs learning with large language models (llms). *arXiv preprint arXiv:2310.09872*, 2023.

[130] Frederik Wenkel, Guy Wolf, and Boris Knyazev. Pretrained language models to solve graph tasks in natural language. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023.

[131] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. *arXiv preprint arXiv:2310.13023*, 2023.

[132] Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Yijian Qin, Simin Wu, and Wenwu Zhu. Llm4dyg: Can large language models solve problems on dynamic graphs? *arXiv preprint arXiv:2310.17110*, 2023.

[133] Yijian Qin, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Disentangled representation learning with large language models for text-attributed graphs. *arXiv preprint arXiv:2310.18152*, 2023.

[134] Kaya Stechly, Matthew Marquez, and Subbarao Kambhampati. Gpt-4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems. *arXiv preprint arXiv:2310.12397*, 2023.

[135] Yijun Tian, Huan Song, Zichen Wang, Haozhu Wang, Ziqing Hu, Fang Wang, Nitesh V. Chawla, and Panpan Xu. Graph neural prompting with large language models. *arXiv preprint arXiv:2309.15427*, 2023.

[136] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061*, 2023.